Sam API Documentation

Max Rottenkolber

Wednesday, 27 July 2016

Table of Contents

1 sam

Tiny, concurrent sampling profiler for Clozure Common Lisp. Sam peeks into the stack of a running process repeatedly to figure out where time is spent.

Example:

```
(sam:profile ()
  (loop for i from 1 to 1e8 sum i))
▷ 39%  CCL::>-2  <no source>
▷ 19%  CCL::FIXNUM-SFLOAT-COMPARE  <no source>
▷ 15%  #<Anonymous Function #x30200B10843F>  <no source>
▷ 12%  SAM:CALL-WITH-SAMPLING  (defun call-with-sampling (fn ...
▷ 11%  CCL::INTEGER-DECODE-SHORT-FLOAT  <no source>
▷  5%  CCL::FIXNUM-DECODE-SHORT-FLOAT  <no source>
```

Credits:

The core functionality of Sam was kindly provided by R. Matthew Emerson.

1.1 call-with-sampling (Function)

Syntax:

— Function: **call-with-sampling** *fn* `&key` *interval*

Arguments and Values:

fn—a *function designator*.

interval—a positive *number* designating a time in seconds. The default is 0.001.

Description:

`call-with-sampling` profiles *fn*, and returns a *hash table* containing the collected samples. *Interval* specifies the time between samples collected.

1.2 profile (Macro)

Syntax:

— Macro: **profile** `(&key` *interval cutoff*`)` `&body` *forms*

Arguments and Values:

interval—a positive *number* designating a time in seconds. The default is 0.001.

cutoff—a *number* of *type* `(real 0 100)` designating a percentage. The default is 3.

forms—*forms* to be evaluated.

Description:

`profile` profiles the evaluation of *forms* and prints a summary of the results, omitting samples that constitute below *cutoff* percent of the total samples. *Interval* specifies the time between samples collected.

1.3 profile-process (Function)

Syntax:

— Function: **profile-process** *process duration* `&key` *interval cutoff*

Arguments and Values:

process—a `process`.

duration—a positive *number* designating a time in seconds.

interval—a positive *number* designating a time in seconds. The default is 0.001.

cutoff—a *number* of *type* `(real 0 100)` designating a percentage. The default is 3.

Description:

`profile-process` profiles *process* for *duration*, and prints a summary of the results, omitting samples that constitute below *cutoff* percent of the total samples. *Interval* specifies the time between samples collected.

1.4 report-samples (Function)

Syntax:

— Function: **report-samples** *h* `&key` *cutoff*

Arguments and Values:

h—a *hash table* containing samples.

cutoff—a *number* of *type* `(real 0 100)` designating a percentage. The default is 3.

Description:

`report-samples` prints a summary of the samples in *h*, omitting samples that constitute below *cutoff* percent of the total samples.

1.5 sample-process (Function)

Syntax:

— Function: **sample-process** *process duration* `&key` *interval*

Arguments and Values:

process—a `process`.

duration—a positive *number* designating a time in seconds.

interval—a positive *number* designating a time in seconds. The default is 0.001.

Description:

`sample-process` profiles *process* for *duration,* and returns a *hash table* containing the collected samples. *Interval* is the time between samples collected.